

CLAIMS

What is claimed is:

- 1 1. A method of providing concurrent access to a resource object, the method comprising
2 the computer-implemented steps of:
3 creating and storing a lock data structure for a particular resource object, the lock data
4 structure comprising data indicative of values for a resource object
5 identification, a lock type, and a version number related to a number of
6 changes to the resource object since the lock data structure was generated;
7 receiving a request from a requesting process for a requested lock type for access to
8 the particular resource object; and
9 determining whether to grant the request based on the requested lock type and the
10 lock type in the lock data structure.
- 1 2. A method as recited in Claim 1, further comprising the step of:
2 if it is determined to grant the request, then
3 bringing the value of the lock type in the data structure into agreement with
4 the lock type in the request;
5 generating a lock object having data indicative of the values for the resource
6 object identification, the lock type and the version number from the
7 lock data structure, and
8 returning the lock to the requesting process.
- 1 3. A method as recited in Claim 1, further comprising the steps of:
2 receiving a lock to be released having data indicative of values for the resource object
3 identification and the lock type and the version number;
4 determining whether the data indicative of the value for the lock type in the lock to be
5 released indicates an exclusive lock, and

a
cont

09781525-020301

6 if it is determined the data indicates the exclusive lock is to be released, then changing
7 the value for the version number in the lock data structure based on the value
8 of the version number in the lock to be released.

1 4. A method as recited in Claim 2, wherein:
2 the lock data structure further comprises a reference number;
3 said step of generating a lock data structure further comprises setting the reference
4 number to a predetermined initial value; and
5 said method further comprises, if it is determined to grant the request, then replacing
6 the value of the reference number in the lock data structure with a sum of the
7 value of the reference number in the lock data structure and a predetermined
8 reference change value.

1 5. A method as recited in Claim 4, further comprising the steps of:
2 receiving a lock to be released having data indicating the particular resource object;
3 determining whether the reference number substantially equals the predetermined
4 initial value of the reference number; and
5 if it is determined the reference number does not substantially equal the
6 predetermined initial value, then replacing the value of the reference number
7 in the lock data structure with a difference substantially equal to the value of
8 the reference number in the lock data structure minus the predetermined
9 reference change.

1 6. A method as recited in Claim 5, further comprising, if it is determined the reference
2 substantially equals the predetermined initial value, then deleting the lock data structure for
3 the particular resource object.

1 7. A method of updating a resource object using optimistic locks, the method comprising
2 the computer-implemented step of:

3 receiving from a client process a request to update a particular resource object;
4 sending to a lock manager process a request for a first lock for access to the particular
5 resource object, the request including data indicating an optimistic lock type;
6 receiving the first lock for access to the particular resource object, the first lock
7 including data indicating the resource object, the optimistic lock type and a
8 first value for a version number related to a number of changes to the resource
9 object since the lock manager generated a lock data structure corresponding to
10 the resource object; and
11 using the optimistic lock to update the resource object.

1 8. The method as recited in Claim 7, said using the optimistic lock comprising:
2 sending to a lock manager process a request for a second lock for access to the
3 particular resource object, the request including data indicating the resource
4 object identification and an exclusive lock type;
5 receiving the second lock for access to the particular resource object, the second lock
6 including data indicating the resource object identification, the exclusive lock
7 type and a second value for the version number;
8 determining whether the second value for the version number substantially equals the
9 first value for the version number; and
10 if the second value substantially equals the first value, then
11 committing an updated resource object to the resource, and
12 replacing the second value in the reference number in the second lock with a
13 third value of the version number, the third value computed by adding
14 the second value and a predetermined version change value.

1 9. The method as recited in Claim 8, further comprising, if the second value does not
2 substantially equal the first value, then sending a message to the client process, the message
3 indicating that the resource object was not updated.

1 10. The method as recited in Claim 7, further comprising sending to the lock manager
2 process a first release message to release the first lock.

1 11. The method as recited in Claim 8, further comprising sending to the lock manager
2 process a second release message to release the second lock.

1 12. The method as recited in Claim 9, further comprising sending to the lock manager
2 process a second release message to release the second lock, the second release message
3 including data indicating the third value of the version number in the second lock and the
4 exclusive lock type, wherein the third value of the version number is used by the lock
5 manager to replace the second value of the version number in the lock data structure.

1 13. A computer-readable medium carrying one or more sequences of instructions for
2 providing concurrent access to a resource object, which instructions, when executed by one
3 or more processors, cause the one or more processors to carry out the steps of:
4 generating a lock data structure for a particular resource object, the lock data structure
5 comprising data indicative of values for a resource object identification, a lock
6 type, and a version number related to a number of changes to the resource
7 object since the lock data structure was generated;
8 receiving a request from a requesting process for a requested lock type for access to
9 the particular resource object; and
10 determining whether to grant the request based on the requested lock type and the
11 lock type in the lock data structure.

1 14. A computer-readable medium carrying one or more sequences of instructions for
2 updating a resource object, which instructions, when executed by one or more processors,
3 cause the one or more processors to carry out the steps of:
4 receiving from a client process a request to update a particular resource object;
5 sending to a lock manager process a request for a first lock for access to the particular
6 resource object, the request including data indicating an optimistic lock type;
7 receiving the first lock for access to the particular resource object, the first lock
8 including data indicating the resource object, the optimistic lock type and a
9 first value for a version number related to a number of changes to the resource
10 object since the lock manager generated a lock data structure corresponding to
11 the resource object; and
12 using the optimistic lock to update the resource object.

1 15. An apparatus for providing concurrent access to a resource object, comprising:
2 a processor;
3 one or more stored sequences of instructions which, when executed by the processor,
4 cause the processor to carry out the steps of:
5 generating a lock data structure for a particular resource object, the lock data
6 structure comprising data indicative of values for a resource object
7 identification, a lock type, and a version number related to a number of
8 changes to the resource object since the lock data structure was
9 generated;
10 receiving a request from a requesting process for a requested lock type for
11 access to the particular resource object; and
12 determining whether to grant the request based on the requested lock type and
13 the lock type in the lock data structure.

1 16. An apparatus for updating a resource object, comprising:
2 a processor;

a!
00n¹⁷

50325-0510 (Seq. No. 3256)

3 one or more stored sequences of instructions which, when executed by the processor,
4 cause the processor to carry out the steps of:
5 receiving from a client process a request to update a particular resource object;
6 sending to a lock manager process a request for a first lock for access to the
7 particular resource object, the request including data indicating an
8 optimistic lock type;
9 receiving the first lock for access to the particular resource object, the first
10 lock including data indicating the resource object, the optimistic lock
11 type and a first value for a version number related to a number of
12 changes to the resource object since the lock manager generated a lock
13 data structure corresponding to the resource object; and
14 using the optimistic lock to update the resource object.

1 17. An apparatus for providing concurrent access to a resource object, comprising:
2 means for generating a lock data structure for a particular resource object, the lock
3 data structure comprising data indicative of values for a resource object
4 identification, a lock type, and a version number related to a number of
5 changes to the resource object since the lock data structure was generated;
6 means for receiving a request from a requesting process for a requested lock type for
7 access to the particular resource object; and
8 means for determining whether to grant the request based on the requested lock type
9 and the lock type in the lock data structure.

a'
cont

- 1 18. An apparatus for updating a resource object, comprising:
- 2 a means for receiving from a client process a request to update a particular resource
- 3 object;
- 4 a means for sending to a lock manager process a request for a first lock for access to
- 5 the particular resource object, the request including data indicating an
- 6 optimistic lock type;
- 7 a means for receiving the first lock for access to the particular resource object, the
- 8 first lock including data indicating the resource object, the optimistic lock type
- 9 and a first value for a version number related to a number of changes to the
- 10 resource object since the lock manager generated a lock data structure
- 11 corresponding to the resource object; and
- 12 a means for using the optimistic lock to update the resource object.

09781535-020801